TB1 test

20 Now In class Test
90 min 50%

TB1 Test

J period 50% in exam hall

Part A
Formal language and model of computation

Part B
Limits of computing

Introduction to language

Language is defined in a set of symbols -

Language is a set of symbols alphabet which are combined into acceptable strings

The rules which tell us how to combine strings in a sensible way is called grammar

Alphabet + string + grammar = Language

Alphabet -
A finite, non-empty set of symbols is called an alphabet

Eg $\Sigma = \{a, b, c\}$

String -
A finite sequence of symbols from the alphabet (placed next to each other)

Eg aa, abc, ccb are strings on Σ.

Language -
If Σ is alphabet then a language over Σ is a set of strings whose symbols are

from Σ.

Eg - If $\Sigma = \{a, b\}$ then $L = \{ab, aab, abbbba\}$ is an example of a language over Σ.

If Σ is an alphabet then Σ^* denotes the infinite set of strings that are made up from Σ. Including empty set

- Examples of languages over an alphabet Σ are the sets:

- $\emptyset$, $\{\lambda\}$, Σ, and Σ^*.

Languages are set of strings, so they can be combined by the usual set operations of union and intersection

Product of two languages

If L and M are langs then the new lang called the product of L and M is defined as

$L \cdot \{\lambda\} = L$

$L \cdot \emptyset = \emptyset$

Operation of concatenation is Not commutative. Order does matter

Concatenation is associative.

If L, M and N are languages, then

$$L \cdot (M \cdot N) = (L \cdot M) \cdot N$$

If L is a language, then the product $L \cdot L$ is denoted by L^2.

$$L^0 = \{\lambda\}$$
$$L^n = L \cdot L^{n-1}, \text{ if } n > 0$$

If L is a language over Σ, then the closure of L is the language

denoted by L^* and is defined as follows

$$L^* = L^0 \cup L^1 \cup L^2 \cup \dots$$

The positive closure of L is signified with L^+

$$L^+ = L^1 \cup L^2 \cup L^3 \cup \dots$$

↖ ↑

No empty string

Only difference is empty string

In CS, a language is a set of strings

Grammars

Two collection of symbols are necessary

- Terminals are those symbols which

all strings in language we make

- Non terminals are temporary symbols
  Used to define the grammar replacement rules.
  Must be replaced by terminals.

$$\alpha \to \beta$$

" replace α by β

If $\Sigma = \{a, b\}$ and S is a non-terminal symbol then the rules
$S \to aS$, $S \to \lambda$ are examples of productions for a grammar L.

Example

Grammar is defined by
- set of terminals $T = \{a, b\}$
  Only one terminal start symbol S
  Set of productions

Set of production rules:

$S \to \lambda$, $S \to aSb$

or

$S \to \lambda \mid aSb$

$S \to aSb$ ← Replace S with aSb

$S \to aaSbb$

$S \to aabb$ ⇐ Replace S with λ

Every grammar must have a start symbol.

A string made up from both terminals and nonterminals is called sentential form

Derivation

$a \to B$

$x \alpha y \Rightarrow x \beta y$

$L(G) = \{ s \mid s \in T^* \text{ and } S \Rightarrow^+ s \}$

A production is recursive if its left side occurs on its right side

$S \to aS$ is recursive

Grammar for an infinite language must be directly or indirectly recursive.

Grammar not unique. A language can have many grammars.

Example

$S \to aS \mid bS$

$\therefore$

$L(G) = \emptyset$

because there will always be non-terminals